1.

In this assignment we will revisit an old friend, the traveling salesman problem (TSP). This week you will implement a heuristic for the TSP, rather than an exact algorithm, and as a result will be able to handle much larger problem sizes. Here is a data file describing a TSP instance (original source: http://www.math.uwaterloo.ca/tsp/world/bm33708.tsp).

nn.txt

The first line indicates the number of cities. Each city is a point in the plane, and each subsequent line indicates the x- and y-coordinates of a single city.

The distance between two cities is defined as the Euclidean distance --- that is, two cities at locations (x, y) and (z, w) have distance $\sqrt{(x - z)^2 + (y - w)^2}$ between them.

You should implement the *nearest neighbor* heuristic:

1. Start the tour at the first city.

2. Repeatedly visit the closest city that the tour hasn't visited yet. *In case of a tie, go to the closest city with the lowest index.* For example, if both the third and fifth cities have the same distance from the first city (and are closer than any other city), then the tour should begin by going from the first city to the third city.

3. Once every city has been visited exactly once, return to the first city to complete the tour.

In the box below, enter the cost of the traveling salesman tour computed by the nearest neighbor heuristic for this instance, *rounded down to the nearest integer*.

[Hint: when constructing the tour, you might find it simpler to work with squared Euclidean distances (i.e., the formula above but without the square root) than Euclidean distances. But don't forget to report the length of the tour in terms of standard Euclidean distance.]

Enter answer here